VBA: Generate Elevation Profiles for PolylineM Routes

Contributed by Bert Granberg

03, Nov. 2009
Last Updated 04, Nov. 2009

Calibrated polylineM route features carry a milepost-measure value in addition to the standard x,y coordinate pair.

Having the milepost coordinate makes is convenient for generating the data necessary to make an elevation profile for each route because you can use the m coordinate to make sure all the elevation measures are about the same distance apart.

For the profiles that I generated for Utah highways [access all elevation profiles here], I used the endpoint calibrated routes) together with an SDE-based 10 meter statewide DEM:
- SGID93.TRANSPORTATION.UDOTRoutes_CalibratedEP
- SGID93.RASTER.DEM_10METER

I wanted to do this without having a 3D Analyst license and in an automated manner (i.e. no mouse driving through the ArcMap UI).

The first step was to create a .csv table of the elevation from the DEM at every 1/10th of a mile along the route using a VBS script (below) called: elevationValuesAlongRoutesToCSV

Next I loaded the .csv table into ArcMap and ran a script to automate the creation of the elevation profile graph from the .csv table. For this there are two scripts (below) that offer the choice of a line-based graph or a scatterplot graph.

Although the new graphing functionality (available since the 9.2 release, I think) is very powerful, it's a bit hard to use and never figured out how to control the line graph where 'no data' areas are present in routes with separated parts (like where I-84 runs concurrent with I-15 for a few miles). Graphs can be created with this code for future use within the ArcMap UI or they can be automatically exported to a variety of formats (.png, .pdf, etc.).

Notes:
- The text in the example above was added manually
- Graphs in the linegraphs folder (from the link above) have scaled elevation (you can't judge slope without factoring in the y axis)
- Graphs in the scatterplotsgraphs folder have a non-scaled (constant) elevation representation on the Y axis that is consistent from one graph to the next.

```
Public Sub elevationValuesAlongRoutesToCSV()

    'AGRC-BG 11/3/09
    'this takes a long time...better to run over night

    Dim outputDir As String
    outputDir = "c:/temp"

    Dim pMxDoc As IMxDocument
    Dim pRasterLayer As IRasterLayer
    Dim pRouteFL As IFeatureLayer
    Dim pRouteFC As IFeatureClass
    Dim pRouteFeatureCursor As IFeatureCursor
    Dim pRouteFeature As IFeature
    Dim pPointCollection As IPointCollection
    Dim P As Long
    Dim pVertexPoint As IPoint
    Dim pQF As IQueryFilter
    Set pQF = New QueryFilter
    Set pMxDoc = ThisDocument

    'SET THESE LAYER INDEX NUMBERS
    'PolylineM layer of routes
    Set pRouteFL = pMxDoc.FocusMap.Layer(0)
    'Raster Elevation Model (Grid-based DEM)
    Set pRasterLayer = pMxDoc.FocusMap.Layer(1)

    Set pRouteFC = pRouteFL.FeatureClass

    Dim pIdentify As IIdentify
    Set pIdentify = pRasterLayer

    Set pRouteFeatureCursor = pRouteFC.Search(pQF, True)
    Set pRouteFeature = pRouteFeatureCursor.NextFeature

    Dim response As String
    Dim responseConv As Double
    Dim pPolyCurve As IPolycurve
    Dim labelStr As String
    Dim mCoord, zCoord As Double
    Dim pArray As IArray
    Dim pRasObj As IRasterIdentifyObj

    Open outputDir & "/elevoutfile_all.csv" For Output As #1
    Print #1, "RT_LABEL, C_ID, MCOORD, ZCOORD"

    Do Until pRouteFeature Is Nothing

        labelStr = pRouteFeature.value(pRouteFeature.Fields.FindField("LABEL"))
        Set pPolyCurve = pRouteFeature.Shape

        pPolyCurve.Generalize 2
        pPolyCurve.Densify 160.9, 0
        Set pPointCollection = pPolyCurve

        For P = 0 To pPointCollection.PointCount - 1

            Set pVertexPoint = pPointCollection.Point(P)
            If (pVertexPoint.M >= 0.1) Then

                Set pArray = pIdentify.Identify(pVertexPoint)
                Set pRasObj = pArray.Element(0)
                responseConv = CLng(pRasObj.Name) * 3.2808
                Debug.Print labelStr & "," & P & "," & pVertexPoint.M & "," & responseConv

                Print #1, labelStr & "," & P & "," & pVertexPoint.M & "," & responseConv

            End If
        Next P

        Set pRouteFeature = pRouteFeatureCursor.NextFeature

    Loop
    Close #1

End Sub

Private Sub makeElevationProfile_LineGraph()

    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' get the name of the layer containing feature points
    Dim pTable As ITable
    Dim pTC As ITableCollection
    Set pTC = pMxDoc.FocusMap

    ' SET THIS
    'table containing all points along route
    ' for which z values have been derived
    Set pTable = pTC.Table(0)

    Dim routeListStr As String
    Dim routeList() As String
    Dim r As Long
    Dim qStr As String
    Dim rtStr As String

    'OPTION #1
    'do all routes in positive direction
    qStr = "not (RT_LABEL like '%X' or RT_LABEL like '%N')"
    'OR do a single route
'    qStr = "RT_LABEL like '0153P'" 'to do a specific route

    routeListStr = getUniqueValues(pTable, "RT_LABEL", qStr)
    routeList = Split(routeListStr, ",")


    For r = 0 To UBound(routeList)

        rtStr = routeList(r)
        Debug.Print rtStr

        ' create graph
        Dim pDataGraphBase As IDataGraphBase
        Dim pDataGraphT As IDataGraphT
        Set pDataGraphBase = New DataGraphT
        Set pDataGraphT = pDataGraphBase

        ' graph title
        pDataGraphT.GeneralProperties.Title = rtStr & " Elevation Profile Graph"
        pDataGraphBase.Name = "Graph of " & rtStr

        ' create vertical line series & customize graph color, symbol, axis labels
        Dim pSP As ISeriesProperties
        Dim pDGTPenProperties As IDataGraphTPenProperties
        Dim pLineSeriesProperties As ILineSeriesProperties
        Dim pDataGraphTSymbolProperties As IDataGraphTSymbolProperties
        Dim pDataGraphTAxisProperties As IDataGraphTAxisProperties

        Set pSP = pDataGraphT.AddSeries("line:vertical")
        pSP.colorType = esriGraphColorMatch
        pSP.WhereClause = "RT_LABEL = '" & rtStr & "'"
        pSP.InLegend = False

        pSP.SourceData = pTable
        pSP.SetField 0, "Mcoord"
        pSP.SetField 1, "ZCoord"

        Set pLineSeriesProperties = pSP
        Set pDataGraphTSymbolProperties = pLineSeriesProperties.SymbolProperties
        pDataGraphTSymbolProperties.Style = esriDataGraphTSymbolNothing
        Set pDGTPenProperties = pSP.PenProperties
        pDGTPenProperties.Style = esriDataGraphTPenSolid
        pDGTPenProperties.Color = RGB(204, 51, 51)

        Set pDataGraphTAxisProperties = pDataGraphT.AxisProperties(0)
        pDataGraphTAxisProperties.Title = "Approx. Elevation (feet)"
        pDataGraphTAxisProperties.Maximum = 12000 'this doesn't seem to work
        pDataGraphTAxisProperties.Minimum = 2000 'this doesn't seem to work

        Set pDataGraphTAxisProperties = pDataGraphT.AxisProperties(2)
        pDataGraphTAxisProperties.Title = "LRS Milepost"

        Dim pSortFlds As IDataSortSeriesProperties
        Set pSortFlds = pSP
        Dim idx As Long
        pSortFlds.AddSortingField "MCoord", True, idx

        Dim pCancelTracker As ITrackCancel
        Set pCancelTracker = New CancelTracker
        pDataGraphT.Update pCancelTracker


        'OPTION #2 OUTPUT
        'VIEW IN ARCMAP OPTION
        ' create data graph window within ArcMap
        ' uncomment this to produce graphs within ArcMpa
'       Dim pDGWin As IDataGraphWindow2
'       Set pDGWin = New DataGraphWindow
'       Set pDGWin.DataGraphBase = pDataGraphBase
'       Set pDGWin.Application = ThisDocument.Parent
'       pDGWin.Show (True)
'
'       Dim pDataGraphs As IDataGraphCollection
'       Set pDataGraphs = pMxDoc
'       pDataGraphs.AddDataGraph pDataGraphBase

        'EXPORT
        ' export the graph instead of displaying in ArcMap
        ' use the following code and comment the above 9 lines

        Dim fileName As String
        fileName = "c:\temp\" & rtStr & ".png"
        pDataGraphT.ExportToFileEx fileName, 800, 500
        fileName = "c:\temp\" & rtStr & ".pdf"
        pDataGraphT.ExportToFileEx fileName, 800, 500

    Next r

End Sub


Private Sub makeElevationProfile_ScatterGraph()

    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    ' get the name of the layer containing feature points
    Dim pTable As ITable
    Dim pTC As ITableCollection
    Set pTC = pMxDoc.FocusMap

    ' SET THIS
    'table containing all points along route
    ' for which z values have been derived
    Set pTable = pTC.Table(0)

    Dim routeListStr As String
    Dim routeList() As String
    Dim rtStr As String
    Dim r As Long
    Dim qStr As String
    'OPTION #1
    'do all routes in positive direction
    qStr = "not (RT_LABEL like '%X' or RT_LABEL like '%N')"
    'OR do a single route
'    qStr = "RT_LABEL like '0153P'" 'to do a specific route

    routeListStr = getUniqueValues(pTable, "RT_LABEL", qStr)
    routeList = Split(routeListStr, ",")


    For r = 0 To UBound(routeList)

        rtStr = routeList(r)
        Debug.Print rtStr

        ' create graph
        Dim pDataGraphBase As IDataGraphBase
        Dim pDataGraphT As IDataGraphT
        Set pDataGraphBase = New DataGraphT
        Set pDataGraphT = pDataGraphBase

        ' graph title
        pDataGraphT.GeneralProperties.Title = rtStr & " Elevation Profile Graph"
        pDataGraphBase.Name = "Graph of " & rtStr

        ' create vertical line series & customize graph color, symbol, axis labels
        Dim pSP As ISeriesProperties
        Dim pDGTPenProperties As IDataGraphTPenProperties
        Dim pPointSeriesProperties As IPointSeriesProperties
        Dim pDataGraphTSymbolProperties As IDataGraphTSymbolProperties
        Dim pDataGraphTAxisProperties As IDataGraphTAxisProperties

        Set pSP = pDataGraphT.AddSeries("scatter_plot")
        pSP.colorType = esriGraphColorMatch
        pSP.WhereClause = "RT_LABEL = '" & rtStr & "'"
        pSP.InLegend = False

        pSP.SourceData = pTable
        pSP.SetField 0, "Mcoord"
        pSP.SetField 1, "ZCoord"

        Set pPointSeriesProperties = pSP
        Set pDataGraphTSymbolProperties = pPointSeriesProperties.SymbolProperties
        pDataGraphTSymbolProperties.Style = esriDataGraphTSymbolCircle

        pDataGraphTSymbolProperties.Color = RGB(204, 51, 51)
        pDataGraphTSymbolProperties.Width = 1
        Set pDGTPenProperties = pSP.PenProperties
        pDGTPenProperties.Style = esriDataGraphTPenSolid

        Set pDataGraphTAxisProperties = pDataGraphT.AxisProperties(0)
        pDataGraphTAxisProperties.Title = "Approx. Elevation (feet)"
        pDataGraphTAxisProperties.AutomaticMaximum = False
        pDataGraphTAxisProperties.AutomaticMinimum = False
        pDataGraphTAxisProperties.Maximum = 10500
        pDataGraphTAxisProperties.Minimum = 2000

        Set pDataGraphTAxisProperties = pDataGraphT.AxisProperties(2)
        pDataGraphTAxisProperties.Title = "LRS Milepost"

        Dim pCancelTracker As ITrackCancel
        Set pCancelTracker = New CancelTracker
        pDataGraphT.Update pCancelTracker
        'OPTION #2 OUTPUT
        'VIEW IN ARCMAP OPTION
        ' create data graph window within ArcMap
        ' uncomment this to produce graphs within ArcMpa
'        Dim pDGWin As IDataGraphWindow2
'        Set pDGWin = New DataGraphWindow
'        Set pDGWin.DataGraphBase = pDataGraphBase
'        Set pDGWin.Application = ThisDocument.Parent
'        pDGWin.Show (True)
'
'        Dim pDataGraphs As IDataGraphCollection
'        Set pDataGraphs = pMxDoc
'        pDataGraphs.AddDataGraph pDataGraphBase

        'EXPORT
        ' export the graph instead of displaying in ArcMap
        ' use the following code and comment the above 9 lines

        Dim fileName As String
        fileName = "c:\temp\" & rtStr & ".png"
        pDataGraphT.ExportToFileEx fileName, 800, 500
        fileName = "c:\temp\" & rtStr & ".pdf"
        pDataGraphT.ExportToFileEx fileName, 800, 500

    Next r

End Sub
Public Function getUniqueValues(inTable As ITable, sFieldName As String, queryStr As String) As String

      Dim pData As esriGeoDatabase.IDataStatistics
      Dim pCursor As esriGeoDatabase.ICursor
      Dim pStatResults As esriSystem.IStatisticsResults
      Dim pQF As IQueryFilter
      Set pQF = New QueryFilter
      pQF.WhereClause = queryStr

      Set pCursor = inTable.Search(pQF, False)

      Set pData = New esriGeoDatabase.DataStatistics
      pData.Field = sFieldName
      Set pData.Cursor = pCursor

      Dim pEnumVar As esriSystem.IEnumVariantSimple, value As Variant
      Dim iCnt As Integer
      iCnt = 0
      Set pEnumVar = pData.UniqueValues
      value = pEnumVar.Next

      Do Until IsEmpty(value)
        If iCnt = 0 Then
            getUniqueValues = value
        Else
            getUniqueValues = getUniqueValues + "," & value
        End If
        value = pEnumVar.Next
        iCnt = iCnt + 1
      Loop
End Function
```